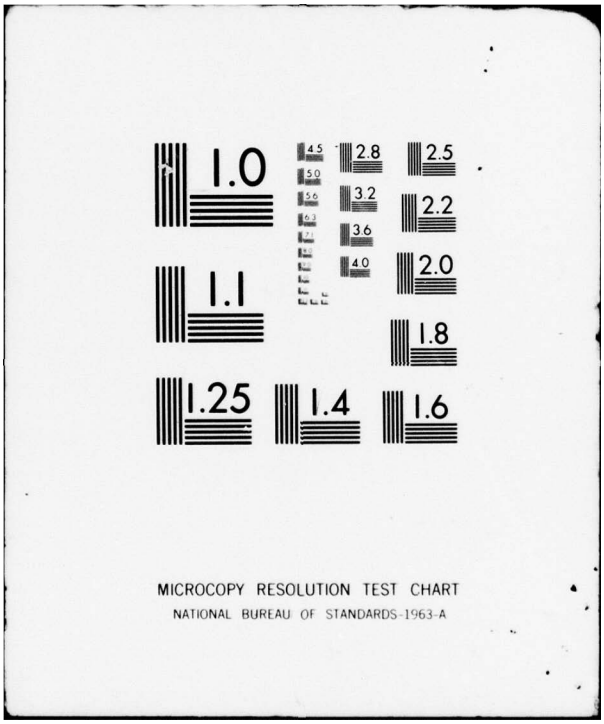
AD-A056 208 NATIONAL MILITARY COMMAND SYSTEM SUPPORT CENTER WASH--ETC F/G 9/2
NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (N--ETC(U)
JUN 75

UNCLASSIFIED NMCSSC-CSM-UH-15-74-V4-1/ NL

END
DATE FILMED
8 -78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DEFENSE COMMUNICATIONS AGENCY
NATIONAL MILITARY COMMAND SYSTEM
SUPPORT CENTER
WASHINGTON, D. C. 20301

NMCS Information Processing System 360 Formatted File System (NIPS 360 FFS). Users Manual. Volume IV. Retrieval and Sort Processor (RASP). Changes 1, 2, and 3.

1 June 1975

LEVEL

IN REPLY REFER TO:

TO: DISTRIBUTION

AD-A002948

SUBJECT: Change 1 to CSM UM 15-74, Retrieval and Sort Processor (RASP), Volume IV, dated 15 October 1974

NMCSSC-CSM-UM-15-74-V4-1/2/3

1. Insert the enclosed change pages and destroy the replaced pages according to applicable security regulations.

42 p.

2. A list of Effective Pages to verify the accuracy of this manual is enclosed. This list should be inserted before the title page.

3. When this change has been posted, make an entry in the Record of Changes on the inside cover.

FOR THE COMMANDER:

6 Enclosures
Change 1 pages

J. DOUGLAS POTTER
Chief, Military Personnel and Administrative Services Office

243 000

JCB

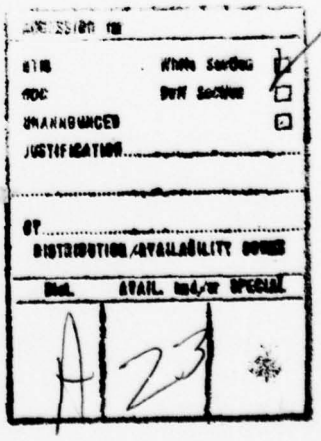
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

EFFECTIVE PAGES - 1 June 1975

This list is used to verify the accuracy of CSM UM 15-74, Volume IV, after change pages have been inserted. Original pages are indicated by the letter O, change 1 by the numeral 1.

Page No.	Change No.
TITLE	O
ii	O
iii - iv	1
v - 116	O
117 - 120	1
121 - 132	O



CONTENTS

4.3 Transaction Confirmation

The source-form of all retrievals in a job is listed on the on-line printer to provide a permanent record of the retrievals submitted in each batch. Errors detected during the edit pass are flagged and error codes are printed in the left margin opposite the invalid statement.

4.4 File Indexing Statistics and Messages

Errors detected by the file indexing function are communicated to the user by an error code and text as appropriate. As indexing operates on retrievals that are essentially error-free, the retrieval statements will not be printed out. The clauses that activate index usage (i.e., the must-satisfy clauses referencing indexed fields) are printed.

Index Processing prints the number of candidate records that will be examined by Retrieval Proper to determine the qualifying records.

4.5 File Analysis and Run Optimization Statistics

The File Analysis Statistics Capability in the RASP component provides transactions for each query executed in a RASP execution. The DSNAME of this data set must be the data file name plus a T. The T is added to ISAM names; the S is replaced by T in SAM names. To obtain transaction output, the DSNAME must be cataloged and the user must specify the volume serial (VTRANS) and unit (UTRANS) in the execution procedure. The volume may be any direct access volume. If the data set exists at execution time, transactions will be added (DISP=MOD). If it does not exist, five tracks will be dynamically allocated. The user may change this value by overriding the TRANST DD card space parameter. Transactions are written as fixed length, unblocked, 50-byte records. The format (fixed) and length (50) cannot be changed but the user may change the blocking factor by specifying a DCB BLKSIZE in the TRANST DD card which is a multiple of 50.

If the user specifies a DSNAME (TRANS) in the TRANS DD card, he must supply all parameters required to process the data set. These parameters must conform to the requirements defined here.

The Run Optimization Statistics capability provides the user with statistical data reflecting the core allocation during a RASP execution. The breakdown of the statistics details the amount of core used for user subroutines and tables, queries, stash area, I/O buffers, and access methods. It also includes the number of BLDL entries allocated and used and the number of entries required for all subroutines, tables, and the query to reside in core. The amount of core required for each subroutine and table and the query to reside in core will also be output. If subroutines and tables are rolled, this information will be output with the causes for the rolling and the number of times it occurred.

The user is able to enter override parameters for the number of BLDL entries to allocate, and the size of the stash area to be used for storage of data records.

The statistics gathering is initiated through parameters entered in the PARM field of the EXEC card. The parameters and their functions are as follows:

ROS - Gather and output Run Optimization Statistics.

NOROS - Omit statistics. If no other parameter is used, this parameter should be omitted, as it is the default.

The parameters the user may supply to tailor core allocation are as follows:

TCP=nK - The number (n) of 1000 (K) bytes to allocate to stash area.

TCB=n - The number (n) of BLDL entries to allocate.

TCS - Invalid in RASP.

With the above override parameters, run optimization statistics will be gathered and output unless NOROS is also entered.

For a more detailed description of the capability, see Volume I, Introduction to File Concepts.

4.6 Operator Messages

The RASP component is designed to operate with minimum operator intervention. All operator-system communication is performed under control of the Operating System.

Section 5

JOB MANAGEMENT

CSM UM 15-74, Volume VIII, Job Preparation Manual, illustrates the use of the cataloged procedure XRASP. It may be used for single file retrievals or for merged-file retrievals. The "symbolic parameter" capability provided by the OS/360 is utilized by the procedure. This provides the user with a simple method of designating the data sets to be used for a given run.

A standard set of defaults have been specified for the symbolic parameters. Assuming that all data file and library data sets have been cataloged for the typical run, only the data file and library require naming at run time. Default parameters, a listing of the XRASP procedure and DD name usage are provided in the Job Preparation Manual.

The procedure has been defined to suffix the names of sequential data files with "S," Index Data Sets with "X," and user libraries with "L." This convention is standard for all components in NIPS 360 PPS. These suffix characters must be taken into consideration when cataloging the data sets and when using them at execution time.

5.1 Checkpoint/Restart

During SAM processing, the user may invoke the CS/360 checkpoint/restart capability to record timed or end-of-volume checkpoints. If checkpoints are needed during ISAM processing, only the timed option is valid. The checkpoint/restart capability should only be used during long-running jobs using the execute only procedures. (Note that the CS/360 step restart is program independent and is not the topic of this discussion.) A detailed description of the CS/360 check/restart capability (which is utilized in

DEFENSE COMMUNICATIONS AGENCY
COMMAND AND CONTROL
TECHNICAL CENTER
WASHINGTON, D. C. 20301

10 June 1976

IN REPLY
REFER TO:

TO: DISTRIBUTION

SUBJECT: Change 2 to CSM UM 15-74, Retrieval and Sort Processor (RASP), Volume IV, dated 15 October 1974

1. Insert the enclosed change pages and destroy the replaced pages according to applicable security regulations.

2. A list of Effective Pages to verify the accuracy of this manual is enclosed. This list should be inserted before the title page.

3. When this change has been posted, make an entry in the Record of Changes on the inside cover.

FOR THE DIRECTOR:

J. DOUGLAS POTTER
Assistant to the Director
for Administration

14 Enclosures
Change 2 pages

EFFECTIVE PAGES - 10 June 1976

This list is used to verify the accuracy of CSM UM 15-74, Volume IV, after change 2 pages have been inserted. Original pages are indicated by the letter O, change 2 by the numeral 2.

Page No.	Change No.
Title Page	2
ii	2
iii - iv	2
1 - 66	O
67 - 70	2
71 - 104	O
105 - 106	2
106.1 - 106.2	2
107 - 116	O
117 - 118	2

COMMAND AND CONTROL TECHNICAL CENTER

Computer System Manual Number CSM UM 15-74

10 June 1976

NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (NIPS 360 FFS)

Users Manual

Volume IV - Retrieval and Sort Processor (RASP)

SUBMITTED BY:

C K Hill

CRAIG K. HILL
Captain USA
CCTC Project Officer

APPROVED BY:

Edward Wallace

CEDRIC S. WALLACE
Captain, USN
Deputy Director, NMCS ADP

CH-2

ACKNOWLEDGMENT

This manual was prepared under the direction of the Chief for Programming with general support provided by the International Business Machines Corporation under contracts DCA 100-67-C-0062, DCA 100-69-C-0029, DCA 100-70-C-0031, DCA 100-70-C-0080 and DCA 100-71-C-0047

CONTENTS

effected by the ABSENT condition would cause those answer entry records to be eliminated from the QRT. This situation might arise in a case where more than one clause is specified for an IF statement such as in the following example. (For a description on the SORT statement see section 2.4.3.1.).

```
IF LOC EQ PARIS AND (MEPSD GT 1 OR MEQPT ABSENT).
SORT UIC, MEQPT.
```

In this example it is desired to retrieve those records which have the value PARIS in the field LOC with the further condition that either the number of major equipment processed field is greater than 1 for at least one major equipment subset or there are no major equipment subsets in the record. Any records which qualify as a result of the MEQPT ABSENT clause would be excluded from the answer set since there are no flagged subsets on which to sort those records as requested by the SORT statement.

2.4.2.3.4 FUNCTION Operator

The FUNCTION Operator is a conditional clause within the IF or FURTHER conditional statement. It is normally preceded and followed by one of the logical connectors in the statement. The use of the FUNCTION Operator is designated by the keyword FUNCTION, followed by the name of the function (the user's subroutine which may be enclosed in pound (#) signs). All terms following the function name and preceding the next logical connector or the end of the statement are considered as part of the parameter list to the function subroutine. The parameters may consist of field names, literals and system-provided work areas.

Field names must be designated by prefixing the field name with an ampersand character. Within a parameter list, fields may be referenced which, belong to the fixed set and one periodic set, but not to two different periodic sets.

System-provided work areas will be designated by the system labels WORK1 through WORK5. No distinction is made as to whether a work area is input to or output from the subroutine.

Any item in the parameter list which is not a system work area or is not prefaced with a plus or ampersand character is processed as a literal. Literals containing embedded blanks or commas must be enclosed in quotes, and literals prefaced with a plus or minus sign (+ or -) must also be enclosed in quotes.

Partial-field notation or user subroutine conversion will not be allowed with items in the parameter list.

Example

```
IF...AND FUNCTION GCD,+POINT,6Ø75N12323E,9ØØ,GT,
WORK1 CR...
```

Comment: Assume that GCD is a user-written subroutine which, given two latitude/longitude points, will compute the distance between the two points and compare it to a given distance. It then will set the true/false indicator according to the given condition. The computed distance will be placed in a designated work area. The order of parameters is Point 1, Point 2, distance, qualifier and work area.

In this example, Point 1 is a data file field POINT. Point 2 is a user-designated point 6Ø75N12535E. Note that for coordinates in a parameter list, conversion to internal form is not automatic. The designated distance is 9ØØ and to qualify, the computed distance must satisfy the condition GT. Note here that GT is a requirement of the subroutine GCD. To RASP, GT is a literal value to be entered as a parameter to GCD. The computed distance would be placed in the system-provided work area WORK1.

RASP only provides an interface to the user-written function subroutine. The number of parameters, the order of parameters, whether parameters can be omitted, and invalid parameters (to the subroutine) are all functions of the subroutine. RASP provides the data and control for the subroutine in the order in which they appear in the query. If the subroutine will allow a variable number of parameters, omission within a parameter list cannot be denoted by use of the double comma. To omit a parameter

within the list which is positional in nature, an omit code which is recognized by the subroutine must be used.

There are several rules or restrictions which the user must follow to use the FUNCTION Operator. Basically, the FUNCTION Operator and any correlation between subroutine requirements and designated input are the user's responsibility. If the following rules are heeded, there should be little problem using the FUNCTION Operator.

a. A double comma cannot be used to indicate omission of a parameter.

b. Parameters must be designated in the order prescribed by the subroutine.

c. Literal values in the parameter list are subroutine related and not data field related; thus, there will not be any automatic conversion by subroutines of literals on input.

d. A literal containing embedded blanks or commas, or a literal prefaced with a plus or minus sign, must be enclosed in quotes.

e. A literal containing any nonnumeric character will be passed to the function subroutine as EBCDIC characters; this includes signed numeric values also, because the plus and minus signs are not interpreted as numeric characters.

f. A literal containing all numeric characters will be converted to a binary value prior to being passed to the function subroutine. Enclosure of an all numeric value in quotes will not prevent that value from being converted to binary; i.e., a literal must contain a nonnumeric character to be passed to the function subroutine as an alpha literal.

g. Data file field names must be prefixed with an ampersand.

h. The parameter list to the FUNCTION Operator subroutine cannot contain the word AND or OR.

2.4.2.3.5 FUNCTION Operator Usage by Index Processing Analyzer Routines

The FUNCTION operator parameter string will be examined to determine if a data file field, or fields (a data field name must be prefixed by an ampersand), is present. If so, and any field so referenced is indexed, Index Processing will be invoked. If no field name is given in the FUNCTION operator parameter string, or no field is an index field, index usage will not be possible for that clause. Only one indexed field per function clause (the first one encountered) can trigger index usage. Any other indexed fields are treated as nonindexed fields in the analysis.

The indexed field used in a FUNCTION operator clause must have an analyzer routine specified for it. An analyzer routine is designated for a field in an Index Specification run (reference Volume II, File Structuring). The purpose of the analyzer routine is to analyze the FUNCTION operator parameters for Index Processing. The analyzer routine will return a value (or values) which Index Processing will use as arguments when searching the field's index information in the Index Data Set. This search is conducted in an "equal" relationship only.

Index Processing cannot determine how the FUNCTION operator parameters interrelate; this must be done by the analyzer routine. Therefore, if no analyzer routine is specified for any index field, Index Processing will be negated for the entire FUNCTION operator clause.

Index Processing only provides an interface to the user-written analyzer subroutines, exactly as RASP provides the interface to a Function subroutine. All conditions, rules, and restrictions regarding Function subroutines, as described in Section 2.4.2.2.3, FUNCTION Operator, above, apply to an analyzer subroutine. Section 2.8, Writing an Analyzer Subroutine, provides information for calling sequence and parameters needed.

```
EXECUTE RET1 (LOC=CNAM).
```

The descending sort option would not be applied to the field CNAM, and the results of the retrieval would be an ascending sort on both fields, UIC and CNAM.

2.8 Writing a FUNCTION Operator Subroutine

This subsection is primarily directed towards the programmer responsible for writing the subroutine. The terminology used is programmer-oriented. Based on the qualification algorithm established by the user (analyst) and based on any additional output requirements, the programmer can proceed with detailed design and coding of the subroutine. The subroutine should be written as a single root segment, and no input or output functions should be performed by the user routine.

The Function Operator subroutine should be compiled and link edited onto the Partitioned Data Set Library, which is used to store program load modules and is included in the SLIB concatenation of data sets when executing RASP. (This library is frequently referred to as the User File Library or Composite Library and contains retrievals, RITs, conversion tables and subroutines.)

The following paragraphs describe the interface or linkage required for Function Operator subroutines. RASP uses the standard NIPS 360 FPS Subroutine Supervisor (SUBSUP) calling sequence. Input formats can usually be predetermined; thus, the use of COBOL and/or FORTRAN is feasible. More complex and variable input subroutines will require use of OS/360 Assembler Language. The RASP Function Operator capability provides the interface with the user routine to cause execution of the qualification algorithm and to pass generated data to the QRT for display or further processing by the Output Processor.

Passed Data String Format

RASP will "gather" data elements of the Function Operator parameter list as prescribed by the user in his query and will formulate a data string which will be passed

to the named subroutine. In this string, there may be elements from the data file, from literals within the parameter list or from the system work areas. All numeric data from the data file and system work areas, and from unsigned numeric literals, will be passed as fullword binary values.

Signed numeric literals are passed as EBCDIC literals and must be converted to the proper form by the subroutine. The maximum or minimum value passed as a fullword binary value is ±2147483647. The passed data string format is as follows:

AABBCCDEFF..FFCCDEFF..FF....

where

A = Number of bytes in the data string

B = Number of data elements in the data string

C = Number of bytes (length) of the data element

D = Type of data element code

A = Alpha (decimal)

B = Binary

C = Coordinate

E = Set code (Ø = fixed set, 1 = periodic set, 2 = work area, 3 = literal)

F = Data value.

C, D, E, and F are repeated for each item in the parameter list. If a parameter is replaced by use of the replacement capability, the length and mode for the replaced value will be used. Obviously, if the replacement length is different from the original length, format problems will be encountered when predefined fixed formats are used. Note also that a $ mask replacement value will result in a data

element length of zero, and that mode and set code will not be set.

Alignment will not be guaranteed; however, if all numeric values appear first in the parameter list, they will be fullword aligned.

4.3 Transaction Confirmation

The source-form of all retrievals in a job is listed on the on-line printer to provide a permanent record of the retrievals submitted in each batch. Errors detected during the edit pass are flagged and error codes are printed in the left margin opposite the invalid statement.

4.4 File Indexing Statistics and Messages

Errors detected by the file indexing function are communicated to the user by an error code and text as appropriate. As indexing operates on retrievals that are essentially error-free, the retrieval statements will not be printed out. The clauses that activate index usage (i.e., the must-satisfy clauses referencing indexed fields) are printed.

Index Processing prints the number of candidate records that will be examined by Retrieval Proper to determine the qualifying records.

4.5 File Analysis and Run Optimization Statistics

The File Analysis Statistics Capability in the RASP component provides transactions for each query executed in a RASP execution. The data set name (DSNAME) of this data set must be the data file name suffixed by a T. The T is added to ISAM names; the S is replaced by T in SAM names. To obtain transaction output, the DSNAME must be cataloged and the user must specify the volume serial (VTRANS) and unit (UTRANS) in the execution procedure. The volume may be any direct access volume.

If the transaction data set exists at execution time, transactions will be added (DISP=MOD). If it does not exist, a five track data set will be dynamically allocated. The user may change the allocation value by overriding the TRANST DD card space parameter. Transactions are written as fixed length, unblocked, 50-byte records. The format (fixed) and length (50) cannot be changed but the user may change the blocking factor by specifying a DCB BLKSIZE in the TRANST DD card which is a multiple of 50.

If the user specifies a DSNAME (TRANS) in the TRANS DD card, he must supply all parameters required to process the data set. These parameters must conform to the requirements defined above

The Run Optimization Statistics capability provides the user with statistical data reflecting the core allocation during a RASP execution. The breakdown of the statistics details the amount of core used for user subroutines and tables, queries, stash area, I/O buffers, and access methods. It also includes the number of BIDL entries allocated and used and the number of entries required for all subroutines, tables, and the query to reside in core. The amount of core required for each subroutine and table and the query to reside in core will also be output. If subroutines and tables are rolled, this information will be output with the causes for the rolling and the number of times it occurred.

The user is able to enter override parameters for the number of BIDL entries to allocate, and the size of the stash area to be used for storage of data records.

The statistics gathering is initiated through parameters entered in the PARM field of the EXEC card. The parameters and their functions are as follows:

ROS - Gather and output Run Optimization Statistics.

NOROS - Omit statistics. If no other parameter is used, this parameter should be omitted, as it is the default.

The parameters the user may supply to tailor core allocation are as follows:

TCP=nK - The number (n) of 1000 (K) bytes to allocate to stash area.

DEFENSE COMMUNICATIONS AGENCY
COMMAND AND CONTROL
TECHINCAL CENTER
WASHINGTON, D.C. 20301

IN REPLY
REFER TO:

15 October 1977

TO: DISTRIBUTION

SUBJECT: Change 3 to CSM UM Retrieval and Sort Processor (RASP), Volume IV, dated 15 October 1974.

1. Insert the enclosed change pages and destroy the replaced pages according to applicable security regulations.

2. A list of Effective Pages to verify the accuracy of this manual is enclosed. This list should be inserted before the title page.

3. When this change has been posted, make an entry in the Record of Changes on the inside cover.

FOR THE DIRECTOR:

18 Enclosures
Change 3 pages

J. DOUGLAS POTTER
Assistant to the Director
for Administration

EFFECTIVE PAGES - 30 September 1977

This list is used to verify the accuracy of CSM UM 15-74, Volume IV, after change 3 pages have been inserted. Original pages are indicated by the letter O, change 3 by the numeral 3.

Page No.	Change No.
Title Page	O
ii	O
iii-iv	3
v-2	O
3-6	3
7-66	O
67-68	3
69-112	O
112.1-112.4	3
113-118	3
119-132	O

CONTENTS

j. Dynamic analysis of retrieval logic to determine if index processing is applicable. If so, RASP will restrict the detailed logic examination to the potential qualifying records as indicated by Index Processing.

k. Selection of only desired periodic sets with the fixed set rather than whole qualifying logical blocks

l. The ability to use user-written subroutines to qualify data through use of the Function Operator

m. The ability to pass information generated by the user-written subroutines to OP through use of system-provided work areas

n. The ability to change operands at execution time through use of the skeleton query feature

o. The ability to change sort field designations at execution time through use of the skeleton query feature

Retrieval specifications may be from card input or from specifications stored as executable modules on the system or a file library. In addition, the Terminal Processing Component uses RASP to process queries from a display console.

RASP will retrieve from data files organized sequentially on magnetic tape or as an indexed sequential data set resident on a Direct Access Storage Device (DASD). RASP will also retrieve data from a Virtual Storage Access Method (VSAM) file.

1.1 Secondary Indexing

Secondary Indexing provides the user the capability to index a data file by the contents of a field other than the Record ID. The primary purpose of the capability is to provide a faster response time for qualifying data records

during retrieval. The principles of Secondary Indexing are described in Volume I, Introduction to File Concepts.

The decision whether file indexing information is actually usable is made on the basis of user-supplied retrieval logic at run time. RASP determines if an Index Data Set is associated with the data file based on the existence of Index Descriptor Records in the data file. If there are none, RASP proceeds with its regular functions of retrieving records.

If indexes have been specified for the file, RASP will invoke the Index Processing function to determine if indexing can be utilized or whether the entire data file must be examined. Index Processing decides this based on the logical criteria for data selection as provided by the IF statements. The format of this statement and the criteria used in determining if indexing can be utilized are described in section 2.4.2.2.

Index Processing returns to RASP with an indication of whether the entire data file must be searched, or whether, based on the user logic, only certain records could possibly qualify. The keys of these records are passed to Retrieval Proper and used to access the records and examine them in detail. By restricting the search to just these records that could qualify, RASP can save the time otherwise required to access and examine records that could not possibly qualify.

1.2 Keyword Indexing

The Keyword capability is a text-retrieval capability that provides a method by which records can be accessed and retrieved based upon the known contents of variable length or text data fields. Just as secondary indexing allows access of only those records known to contain the fixed length fields of interest, keyword indexing allows retrieval of records based on the presence of "keywords" within a field. The selection of records based on these "keywords" is accomplished via a KEYWORD statement which is included in the query. It is described in paragraph 2.4.2.2. The

principles of keyword indexing are described in Volume I, Introduction to File Concepts.

Prior to employing keyword retrieval, the fields to be examined must have been defined as keyword indexed fields either through File Structuring or the Index Specification utility (see Volume II, File Structuring, Index Definition for Keyword Fields for the methods of defining keyword indexed fields). Depending on when the indexes were specified, either File Maintenance or the utility will maintain the index data set with the fields and their designated keyword values. (See Volume I, Introduction to File Concepts, Keyword Indexing Capability for keyword qualification requirements.) The user determines what values in the field qualify as keywords and from this selection bases the arguments for his retrieval.

Keyword processing is initiated by including a KEYWORD statement in the query. The KEYWORD statement operates on the same level as the secondary LIMIT statement. Only one KEYWORD statement per query is allowed. Structurally and functionally, the statement is similar to the IF statement. Qualification is at the record level; that is, subsets are not flagged although qualification may be the result interrogation of periodic data.

If keyword qualification produces a candidate list of qualifying records but secondary indexing is found unfeasible due to a low percentage of fields in the query being indexed, the keyword list becomes the qualifying list for the query. Otherwise, the list of qualifying records from the KEYWORD statement is merged with the list of qualifying records from the secondary indexed fields in the IF statement to produce one final list containing only those record IDs that contain both qualifying keyword and secondary indexed fields. This list is passed to the retrieval routine to indicate what records are to be accessed and retrieved.

If either source produces no candidates, the retrieval is terminated with an advisory message.

1.3 System Flow

RASP has three sections: the Input Processor section, the Retrieval Proper section, and the Sort section. The flow within each section is discussed briefly in the following paragraphs.

The Input Processor section reads the input stream. It edits each statement and restructures communication records for the rest of the component. It extracts the required information from the File Format Table, resolves all field name addresses and references, and places those data values converted by conversion subroutines in the communication records. It also applies an algorithm to restructure the conditional logic into a bit pattern which can be more readily interrogated using programmed logic. Finally, it converts the retrieval statements into macros which are assembled and link-edited by OS/360 to produce an executable load module. The resulting executable retrieval is stored on the Permanent or Temporary Retrieval Library.

The Retrieval Proper section reads the communication records and calls and executes the code stored on the Permanent or Temporary Retrieval Library. It checks to see if the retrieval, the required subroutines, and the first data set to be retrieved against are available.

If either of the first two are not available, an advisory message is issued, and the job is terminated. If the data set is not available, the operator is notified, and appropriate operator action must be taken.

During initialization, Retrieval Proper determines whether an Index Data Set is associated with the file. If so, it calls the Index Processing function which will determine if indexing information can be utilized. If so, a list of potential qualifiers, called candidates, will be presented to Retrieval Proper which will access only those records.

After all initialization and Index Processing functions are completed, Retrieval Proper is read to pass the retrieval logic against data records. Based on the results

effected by the ABSENT condition would cause those answer entry records to be eliminated from the ORT. This situation might arise in a case where more than one clause is specified for an IF statement such as in the following example. (For a description on the SORT statement see section 2.4.3.1.).

```
IF LOC EQ PARIS AND (MEPSD GT 1 OR MEQPT ABSENT).
SORT UIC, MEQPT.
```

In this example it is desired to retrieve those records which have the value PARIS in the field LOC with the further condition that either the number of major equipment processed field is greater than 1 for at least one major equipment subset or there are no major equipment subsets in the record. Any records which qualify as a result of the MEQPT ABSENT clause would be excluded from the answer set since there are no flagged subsets on which to sort those records as requested by the SORT statement.

2.4.2.3.4 FUNCTION Operator

The FUNCTION Operator is a conditional clause within the IF or FURTHER conditional statement. It is normally preceded and followed by one of the logical connectors in the statement. The use of the FUNCTION Operator is designated by the keyword FUNCTION, followed by the name of the function (the user's subroutine which may be enclosed in pound (#) signs). All terms following the function name and preceding the next logical connector or the end of the statement are considered as part of the parameter list to the function subroutine. The parameters may consist of field names, literals and system-provided work areas.

Field names must be designated by prefixing the field name with an ampersand character. Within a parameter list, fields may be referenced which, belong to the fixed set and one periodic set, but not to two different periodic sets.

System-provided work areas will be designated by the system labels WORK1 through WORK5. No distinction is made as to whether a work area is input to or output from the subroutine.

Any item in the parameter list which is not a system work area or is not prefaced with an ampersand character is processed as a literal. Literals containing embedded blanks or commas must be enclosed in quotes, and literals prefaced with a plus or minus sign (+ or -) must also be enclosed in quotes.

Partial-field notation or user subroutine conversion will not be allowed with items in the parameter list.

Example

```
#       IF...AND FUNCTION GCD,&PCINT,6Ø75N12323E,9ØØ,GT,
        WORK1 OR...
```

Comment: Assume that GCD is a user-written subroutine which, given two latitude/longitude points, will compute the distance between the two points and compare it to a given distance. It then will set the true/false indicator according to the given condition. The computed distance will be placed in a designated work area. The order of parameters is Point 1, Point 2, distance, qualifier and work area.

In this example, Point 1 is a data file field POINT. Point 2 is a user-designated point 6Ø75N12535E. Note that for coordinates in a parameter list, conversion to internal form is not automatic. The designated distance is 9ØØ and to qualify, the computed distance must satisfy the condition GT. Note here that GT is a requirement of the subroutine GCD. To RASP, GT is a literal value to be entered as a parameter to GCD. The computed distance would be placed in the system-provided work area WORK1.

RASP only provides an interface to the user-written function subroutine. The number of parameters, the order of parameters, whether parameters can be omitted, and invalid parameters (to the subroutine) are all functions of the subroutine. RASP provides the data and control for the subroutine in the order in which they appear in the query. If the subroutine will allow a variable number of parameters, omission within a parameter list cannot be denoted by use of the double comma. To omit a parameter

#2.10 SYSDATS FUNCTION Subroutine

SYSDATS is a FUNCTION subroutine which allows the user to access the current system date for use in retrieval. By using the subroutine, records can be retrieved based on the logical relation between a file date field and the current system date. The relationship is established by applying an adjustment factor to the system date and then testing this adjusted date against the file data field for the specified relationship.

#2.10.1 SYSDATS Format

The format for using SYSDATS is:

```
FUNCTION SYSDATS relop &fld 'adj1' 'adj2'
```

where:

FUNCTION (required) -- used to identify the string as FUNCTION items.

SYSDATS (required) -- used to identify the FUNCTION subroutine being invoked.

relop (required) -- the relational operator specifying the type of test to be performed. The valid operators are EQ, NE, LT, LE, GT, GE, BT, and NB. Each of these operators is a standard RASP relational operator except for NB, which represents NOT BT.

&fld (required) -- the file fieldname which contains the date value. The field must be a six character ALPHA/DECIMAL field in the form of YYMMDD. The field names must be preceeded by an ampersand (&).

'adj1' (required) -- the adjustment factor to be applied to the system date. The format of the adjustment is:

```
'snnyRnnmonnDA'
```

where:

s (optional) - a sign (+ or -) indicating that the adjustment is to be upward (+) or downward (-). When omitted, the adjustment is assumed to be upward.

nn (required) - a 1 to 2 digit number specifying the number of years, months and/or days the system date is to be adjusted. The number can be any number from zero (0) to 99.

YR - indicates an adjustment in years.

MO - indicates an adjustment in months.

DA - indicates an adjustment in days.

The adjustment factor must be enclosed in single quotes. The adjustment may be specified using one or more of the time units, for example:

'-7DA' - is 7 days prior to the current system date.

'+3MO7DA' - is 3 months and 7 days after the current system date.

'1YR1DA' - is 1 year and 1 day after the current system date, upward adjustment assumed.

If the current system date is to be used without adjustment, the user would code the adjustment as zero (0).

The adjustment factor as applied must not cause the system date to overlap the century.

'adj2' (optional) -- is a second adjustment factor to be applied to the current system date only when either the BT or the NB relational operator is being used. The format 'adj2' is the same as the format of 'adj1'. When

used, the resultant adjusted date must be greater than or equal to the adjusted date derived from 'adj1'.

#2.10.2 SYSDATS Examples

```
IF FUNCTION SYSDATS EQ &FDATE '0'.
```

This example will retrieve records where FDATE is equal to the current system date.

```
OR FUNCTION SYSDATS GT &FDATE '-7DA'.
```

This example will retrieve records where FDATE is greater than 7 days prior to the current system date.

```
AND FUNCTION SYSDATS LE &FDATE '6MO'.
```

This example will retrieve records where FDATE is less than or equal to 6 months after the current system date.

```
FURTHER FUNCTION SYSDATS BT &FDATE '-6DA' '0'.
```

This example will retrieve records where FDATE ranges from 6 days prior to and up to the current system date.

Section 3

INPUT

RASP will retrieve from data files organized sequentially on magnetic tape as single data files or as concatenated segments of a data file, or as either an indexed sequential data set or a virtual storage access data set resident on a Direct Access Storage Device (DASD).

3.1 Data Files

RASP will retrieve from data files organized sequentially on magnetic tape or as either an indexed sequential or a virtual storage access data set resident on a Direct Access Storage Device (DASD). The data set name for the data files may be a qualified data set name up to 44 characters in length. When retrieving from a tape data file, the Queued Sequential Access Method (QSAM) will be used as will the Queued Index Sequential Access Method (QISAM) when retrieving from an ISAM data file or the Virtual Storage Access Method (VSAM) from a VSAM data file residing on a Direct Access Storage Device. The FDT, which describes the logical record structure of the data file, is stored as a part of the data file.

3.2 Index Data Set

This data set is the repository of all indexing information for its associated data file. The data set name is the data file's qualified or unqualified name, less the "S" for SAM files, with the suffix "Y."

3.3 Procedure Library

The Procedure Library contains, among others, a set of Job Control statements for RASP that have been placed in a special data set (SYS1.PROCLIB) and that can be retrieved by

naming it in the OS/360 execute (EXEC) statement. The procedure, XRASP, contains the job and program information, data characteristics and device requirements used by the operating sytem to regulate the execution of RASP job steps, allocate input/output resources, obtain and dispose of data, and communicate with the operator.

3.4 Program Library

The Program Library is a partitioned data set that contains user-written subroutines and tables, retrievals in a program format suitable for loading into main storage for execution, the RASP programs themselves, and standard system-supplied conversion subroutines and tables. The library data set name may be a qualified data set name up to 44 characters in length.

3.5 RASP Control and Source Statements

The RASP Control and Source statements define the environment for the RASP job, provide the logical conditions for one or more retrievals, and specify the library maintenance action to be taken against the Permanent Retrieval Library, a subset of the Program Library.

3.6 Overriding Index Processing

Index Processing normally activates the candidate-access mode if at least 75% of all retrievals in one run can utilize index information. The user can override this by coding an entry on the EXEC statement. The entry is PARM='INDEX=XXXX', where XXXX is one of the following:

NO - Override Index Processing entirely. RASP will not activate Index Processing at all.

EVERY - Allow the candidate-access mode only if all retrievals in a batch can utilize index information.

ANY - Allow the candidate-access mode if any retrieval in the batch can utilize index

information, regardless of the number of universal queries.

However, the presence of a KEYWORD statement in the retrieval negates the PARM override so that indexing is always invoked when the KEYWORD statement is found.

Section 4

OUTPUT

This section provides a general definition of the various forms of output.

4.1 Qualifying Data File (QDF)

The Qualifying Data File (QDF) is written, using the Basic Sequential Access (BSAM), as a sequential file of variable length blocked records on a Direct Access Storage Device in Record ID sequence. Each logical block which satisfies the conditions of one or more retrievals is written on the file. A data record is written only once. The QDF is not sorted and contains no sort keys.

The QDF also includes records which contain file classification, file control data, the FMS logic statements, and source-form retrieval statements for each file.

4.2 Qualifying Record Table (QRT)

The QRT is a sequential file consisting of variable-length blocked records. One of the primary records in this table consists of a sort key and a pointer to the fixed field of a data record on the Qualifying Data File which satisfied the conditions of a retrieval. There is at least one such entry for each record that qualified. Also included in each QRT entry will be information identifying the subsets in each data record which caused the retrieval to be satisfied. There are also records identifying the files used and records identifying the RIT names used.

4.3 Transaction Confirmation

The source-form of all retrievals in a job is listed on the on-line printer to provide a permanent record of the retrievals submitted in each batch. Errors detected during the edit pass are flagged and error codes are printed in the left margin opposite the invalid statement.

4.4 File Indexing Statistics and Messages

Errors detected by the file indexing function are communicated to the user by an error code and text as appropriate. As indexing operates on retrievals that are essentially error-free, the retrieval statements will not be printed out. The clauses that activate index usage (i.e., the must-satisfy clauses referencing indexed fields) are printed.

Index Processing prints the number of candidate records that will be examined by Retrieval Proper to determine the qualifying records.

4.5 File Analysis and Run Optimization Statistics

The File Analysis Statistics Capability in the RASP component provides transactions for each query executed in a RASP execution. The data set name (DSNAME) of this data set must be the data file name suffixed by a T. The T is added to ISAM and VSAM names; the S is replaced by T in SAM names. To obtain transaction output, the DSNAME must be cataloged and the user must specify the volume serial (VTRANS) and unit (UTRANS) in the execution procedure. The volume may be any direct access volume.

If the transaction data set exists at execution time, transactions will be added (DISP=MOD). If it does not exist, a five track data set will be dynamically allocated. The user may change the allocation value by overriding the TRANST DD card space parameter. Transactions are written as fixed length, unblocked, 50-byte records. The format (fixed) and length (50) cannot be changed but the user may change the blocking factor by specifying a DCB BLKSIZE in the TRANST DD card which is a multiple of 50.

If the user specifies a DSNAME (TRANS) in the TRANS DD card, he must supply all parameters required to process the data set. These parameters must conform to the requirements defined above

The Run Optimization Statistics capability provides the user with statistical data reflecting the core allocation during a RASP execution. The breakdown of the statistics details the amount of core used for user subroutines and tables, queries, stash area, I/O buffers, and access methods. It also includes the number of BLDL entries allocated and used and the number of entries required for all subroutines, tables, and the query to reside in core. The amount of core required for each subroutine and table and the query to reside in core will also be output. If subroutines and tables are rolled, this information will be output with the causes for the rolling and the number of times it occurred.

The user is able to enter override parameters for the number of BLDL entries to allocate, and the size of the stash area to be used for storage of data records.

The statistics gathering is initiated through parameters entered in the PARM field of the EXEC card. The parameters and their functions are as follows:

ROS - Gather and output Run Optimization Statistics.

NOROS - Omit statistics. If no other parameter is used, this parameter should be omitted, as it is the default.

The parameters the user may supply to tailor core allocation are as follows:

TCP=nK - The number (n) of 1000 (K) bytes to allocate to stash area.